



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM**

**ĐỀ THI THỬ**  
**MÔN: CẤU TRÚC DỮ LIỆU VÀ**  
**GIẢI THUẬT**  
Thời gian làm bài: 90 phút

**Câu 1:**

1.1

a. Hãy cho biết độ phức tạp của thuật toán Heap sort theo định nghĩa Big-O (O lớn). (0.25 điểm)

b. Hãy cho biết dãy số sẽ thay đổi qua từng bước như thế nào khi áp dụng thuật toán ở câu a, biết rằng dãy số cho như sau: 12, 3, 6, 9 (1 điểm)

1.2 Cho dãy số  $B = \{9, 8, 7, 6, 5, 4, 3, 2, 1\}$ . Hãy cho biết:

a. Những thuật toán sắp xếp nào, trong số các thuật toán insertion sort và selection sort, sẽ gặp thuận lợi khi sắp dãy số B theo thứ tự tăng dần? (0.5 điểm)

b. Thuận lợi đó, theo từng thuật toán đã trả lời ở câu 1.2 a, là gì? (0.5 điểm)

**Câu 2:** An vừa thu thập được n bộ gồm m viên gạch hình vuông được xếp chồng lên nhau thành dạng một tòa tháp theo diện tích giảm dần. Vì trông thấy các tòa tháp có vẻ không đạt chiều cao mong muốn, An muốn kết hợp hết chúng lại để tạo thành một tòa tháp cao hơn. Nhưng vì biết rằng tòa tháp cao và độ cân bằng không cao sẽ rất dễ ngã nên An muốn kết hợp chúng lại trên giấy theo diện tích tăng dần để tính xem tòa tháp có dễ ngã hay không rồi mới thực hiện.

2.1 Hãy mô tả cấu trúc dữ liệu được dùng để lưu diện tích của các viên gạch theo từng bộ riêng biệt. (xem mô tả về input, output ở ví dụ bên dưới)

2.2 Viết chương trình bằng ngôn ngữ C++ in ra diện tích của các viên gạch khi kết hợp n tòa tháp lại theo thứ tự tăng dần, sử dụng cấu trúc dữ liệu ở câu 2.1

Ví dụ input:

- Dòng 1: số nguyên n là số bộ gạch và số nguyên m là số gạch của mỗi bộ.
- n dòng tiếp theo, mỗi dòng gồm m số nguyên dương là diện tích của các viên gạch trong bộ gạch thứ i đã được sắp xếp theo thứ tự tăng dần.

Ví dụ output:

- Một dãy số nguyên dương tăng dần là kết quả của việc ghép n bộ trên.

Input	Output
3 4	2 3 4 5 8 10 16 17 20 25 26 30
3 8 17 20	
2 25 26 30	
4 5 10 16	

**Câu 3:** Cho dãy ký tự như sau: F, D, H, B, A, G, C, E, I Hãy thực hiện các yêu cầu sau:

- Vẽ cây nhị phân tìm kiếm bằng cách thêm lần lượt từng ký tự vào cây theo thứ tự từ trái qua phải của dãy ký tự trên (1 điểm)
- Cho biết kết quả duyệt cây theo RNL, NRL. (0.5 điểm)
- Hủy lần lượt từng nút D, E, F, H trên cây, mỗi lần hủy 1 nút vẽ lại cây nối tiếp theo như thứ tự hủy. (1 điểm)
- Viết hàm đếm số lượng nút có một nút con trên cây, nếu cây rỗng thì in ra giá trị -1. (1 điểm)

**Câu 4:** Cho dãy số sau: 20, 16, 1, 15, 26, 30, 12, 23, 22, 28, 9, 5, 19, 3, 27.

- Hãy vẽ cây B-Tree bậc 3 với từ dãy số trên với thứ tự thêm vào cây từ trái sang phải. Liệt kê các giá trị làm thực hiện thao tác tách nút khi chèn vào.
- Lần lượt xóa các khóa có giá trị 3, 12, 22, 20 khỏi cây trên, vẽ lại cây sau mỗi lần xóa.

**Câu 5:**

5.1 Cho bảng băm T có các đặc điểm sau:

Kích thước bảng băm  $m = 11$ .

Giá trị khóa  $k$  là các số nguyên  $k \in [1, 100]$ .

Sử dụng hàm băm theo phép nhân.

Xử lý đụng độ theo phương pháp thăm dò bậc hai.

Hãy cho biết công thức của hàm băm  $h(k)$  và hàm băm lại  $h_i(k)$ , hay  $h(k, i)$  sao cho xác suất xảy ra đụng độ trong  $T$  không vượt quá  $n/m$  khi  $T$  có  $n$  phần tử. (0.5 điểm)

5.2

Cho  $K$  là một tập các giá trị khóa là số nguyên như sau:  $K=\{89, 18, 10, 12, 49, 58, 69\}$  và bảng băm gồm  $M=10$  ô nhớ trống.

Cho 2 hàm băm như sau:  $h_1(\text{key}) = \text{key} \bmod 10$  và  $h_2(\text{key}) = (\text{key} \bmod 7) + 1$ , trong đó phép toán  $\bmod$  là phép toán lấy phần dư.

Hãy vẽ hình bảng băm khi thêm lần lượt các khóa trong  $K$  vào bảng băm theo thứ tự từ trái qua phải bằng cách dùng hàm băm  $h_1$  để xác định địa chỉ của mỗi khóa. Trong trường hợp xảy ra đụng độ thì dùng phương pháp băm kép (Double Hashing) để giải

quyết đụng độ với hàm băm  $h_i(\text{key}) = (h_1(\text{key}) + i \cdot h_2(\text{key})) \bmod 10$ , trong đó  $i=1,2,3..$  là số lần xảy ra đụng độ của khóa  $\text{key}$  ở lần thứ 1,2,3..

**Câu 6:** Cho đồ thị như hình bên:

a. Xây dựng cấu trúc dữ liệu phù hợp để biểu diễn đồ thị trên.

b. Hãy viết hàm cài đặt giải thuật duyệt đồ thị theo chiều rộng (BFS).

c. Hãy cho biết kết quả duyệt đồ thị theo giải thuật được cài đặt ở câu b và cách biểu diễn ở câu a, với đỉnh xuất phát là đỉnh 0.

